© Elsevier/INRA

Original article

Computation of all eigenvalues of matrices used in restricted maximum likelihood estimation of variance components using sparse matrix techniques

C Robert, V Ducrocq

Station de génétique quantitative et appliquée, Centre de recherches de Jouy-en-Josas, Institut national de la recherche agronomique, 78352 Jouy-en-Josas cedex, France

(Received 16 March 1995; accepted 3 October 1995)

Summary – Restricted maximum likelihood (REML) estimates of variance components have desirable properties but can be very expensive computationally. Large costs result from the need for the repeated inversion of the large coefficient matrix of the mixed-model equations. This paper presents a method based on the computation of all eigenvalues using the Lanczos method, a technique reducing a large sparse symmetric matrix to a tridiagonal form. Dense matrix inversion is not required. It is accurate and not very demanding on storage requirements. The Lanczos method, the computation of eigenvalues, its application in a genetic context, and an example are presented.

Lanczos method / sparse matrix / restricted maximum likelihood / eigenvalue

Résumé – Calcul de toutes les valeurs propres des matrices utilisées dans l'estimation du maximum de vraisemblance restreinte des composantes de variance à l'aide de techniques applicables aux matrices creuses. *Les estimations du maximum de vraisemblance restreinte (REML) des composantes de variance ont des propriétés intéressantes mais peuvent être coûteuses en temps de calcul et en besoin de mémoire. Le problème vient de la nécessité d'inverser de façon répétée la matrice des coefficients des équations du modèle mixte. Cet article présente une méthode basée sur le calcul des valeurs propres et sur l'utilisation de la méthode de Lanczos, une technique permettant de réduire une matrice creuse, symétrique et de grande taille en une matrice tridiagonale. L'inversion de matrices denses n'est pas nécessaire. Cette méthode donne des résultats précis et ne demande que très peu de stockage en mémoire. La méthode de Lanczos, le calcul des valeurs propres, son application dans le contexte génétique et un exemple sont présentés.*

méthode de Lanczos / matrice creuse / maximum de vraisemblance restreinte / valeur propre

INTRODUCTION

The accuracy of estimates of variance components is dependent on the choice of data, method and model. The estimation of (co)variance components by restricted maximum likelihood (REML, Patterson and Thompson, 1971) procedures is generally considered to be the best method for animal breeding data. Furthermore, the animal model is considered to be the model which utilizes the information from the data in the most efficient way. Several different REML algorithms (derivative-free, expectation-maximization, Fisher-scoring) have been used with animal breeding data. Most methods are iterative and require the repeated manipulation of the mixed-model equations (Henderson, 1973).

The derivative-free (DF) algorithm (Smith and Graser, 1986) involves evaluating the log likelihood function explicitly and directly finding the parameters that maximize it. Estimation of (co)variance components does not involve matrix inversion but requires evaluation of:

$$\mathrm{Log}|\mathbf{C}| \qquad [1]$$

at each iteration, where **C** represents the coefficient matrix of the mixed-model equations.

The expectation-maximization (EM) algorithm (Dempster et al, 1977) which utilizes only first derivative information to obtain estimates that maximize the likelihood function of data, requires the diagonal blocks of the inverse of the coefficient matrix for random effects ($\mathbf{C}^{uu}$) and traces of their products with the corresponding inverse of the numerator relationship matrix ($\mathbf{A}^{-1}$). These traces can be written as:

$$\mathrm{tr}[\mathbf{A}^{-1}\mathbf{C}^{uu}] = \mathrm{tr}\left[\mathbf{A}^{-1}(\mathbf{Z}'\mathbf{MZ} + \alpha\mathbf{A}^{-1})^{-1}\right] \qquad [2]$$

where **Z** is the incidence matrix for any random effect, **M** is the absorption matrix of fixed effects and α is the variance ratio. For a comparison of EM- and DF-REML, see Misztal (1994).

The Fisher-scoring (Meyer, 1985) algorithm, which is based on second derivatives of the likelihood function, requires the calculation of more complicated traces like:

$$\mathrm{tr}\left[\mathbf{A}^{-1}\mathbf{C}^{uu}\mathbf{A}^{-1}\mathbf{C}^{uu}\right] = \mathrm{tr}\left[\mathbf{A}^{-1}(\mathbf{Z}'\mathbf{MZ} + \alpha\mathbf{A}^{-1})^{-1}\mathbf{A}^{-1}(\mathbf{Z}'\mathbf{MZ} + \alpha\mathbf{A}^{-1})^{-1}\right] \qquad [3]$$

Expressions in [1–3] have straightforward multitrait extensions. Calculation of the inverses in [2] and [3] is the main computational limitation for the use of REML, particularly when the coefficient matrix is very large. Several attempts have been made to find direct or indirect methods for alleviating these numerical computations. While such algorithms such as DF-REML have proven robust and easy to use, they are generally slow to converge, often requiring many likelihood evaluations, in particular for multivariate analyses fitting several random factors. However, as noted by Graser et al (1987), they only require the factorization of a large matrix rather than its inverse and can be implemented efficiently using sparse matrix techniques for analyses involving several tens of thousands of animals. In the same way, Misztal et al (1993) studied the feasibility of estimating large-scale variance components in an animal model by an EM-REML algorithm using sparse matrix inversion (FSPACK). Other techniques for reducing computational demands based on algorithms using derivatives of the likelihood function (EM or method of scoring procedures) have involved the use of approximations (Boichard et al, 1992) or sampling techniques (Misztal, 1990; Thallman and Taylor, 1991; García-Cortés et al, 1992). Along the same lines, Smith and Graser (1986) have advocated the use of sequences of Householder transformations to reduce the coefficient matrix $\mathbf{C}$ to a tridiagonal form, thus eliminating the need for direct matrix inversion. This is based on the observation that $\text{tr}[\mathbf{C}] = \text{tr}[\mathbf{QCQ}']$ for any $\mathbf{Q}$ such that $\mathbf{QQ}' = \mathbf{Q}'\mathbf{Q} = \mathbf{I}$ and so that $\mathbf{QCQ}'$ is tridiagonal. Furthermore, this idea has been extended by Colleau et al (1989) to compute from the tridiagonalized coefficient matrix the trace of matrix products required in a Fisher-scoring algorithm applied in a multivariate setting and by Ducrocq (1993) with an extra diagonalization step. Diagonalization is the problem of finding eigenvectors and eigenvalues of $\mathbf{C}$. As noted by Dempster et al (1984), Lin (1987) and Harville and Callanan (1990), matrix inversion is avoided if a spectral decomposition of the coefficient matrix is used. Then REML estimation of variance components in mixed models becomes computationally trivial and requires little computer storage. In expressions [1–3], the problem amounts to finding eigenvalues of large sparse symmetric matrices. Indeed [1] can be written in terms of eigenvalues:

$$\log|\mathbf{C}| = \log(\Pi_{i=1}^{n}\lambda_i) = \sum_{i=1}^{n}\log(\lambda_i) \qquad [4]$$

where the λ_i are the eigenvalues of the coefficient matrix $\mathbf{C}$ and, if $\mathbf{L}$ is the Cholesky factor of $\mathbf{A}$ ($\mathbf{A} = \mathbf{LL}'$; Henderson, 1976), [2] and [3] can be simply expressed as:

$$\text{tr}\left[\mathbf{A}^{-1}\mathbf{C}^{uu}\right] = \text{tr}\left[(\mathbf{L}'\mathbf{Z}'\mathbf{MZL} + \alpha\mathbf{I})^{-1}\right] = \sum_{i=1}^{n}[1/(\alpha + \gamma_i)] \qquad [5]$$

$$\text{tr}\left[\mathbf{A}^{-1}\mathbf{C}^{uu}\mathbf{A}^{-1}\mathbf{C}^{uu}\right] = \sum_{i=1}^{n}[1/(\alpha + \gamma_i)^2] \qquad [6]$$

where the γ_i are the eigenvalues of $\mathbf{L}'\mathbf{Z}'\mathbf{MZL}$.

Until now, all of these methods have implied working on dense matrices stored in the core of the computer. These transformations are therefore limited to data sets of moderate size. The purpose of this paper is to present and apply in a genetic context a method for computing some or all eigenvalues of a very large sparse matrix with very little computer storage. This method, attributed to Lanczos (1950), generates a sequence of tridiagonal matrices $\mathbf{T}$, with the property that eigenvalues of $\mathbf{T}_j \in \Re^{j\times j}$ are progressively better estimates of eigenvalues of the large matrix as j increases. This method was developed by Cullum and Willoughby (1985). The computer storage required is minimal.

METHODS

Computing all eigenvalues of a very sparse symmetric matrix

Although the problem of computing eigensystems of symmetric dense matrices that fit easily in the core has been satisfactorily solved using sequences of Householder transformations or Givens rotations (Golub and Van Loan, 1983), the same cannot be said for very large sparse matrices. One way to find some or all eigenvalues of large sparse symmetric matrices **B** is to use the Lanczos method (Lanczos, 1950). If a large matrix is sparse, then the advantages of methods which only use matrix-vector multiplications are obvious as the matrix **B** need not be stored explicitly. All that is needed is a subroutine for efficiently computing the matrix vector product **Bv** for any given vector **v**. In the genetic context considered, it will be seen that such a computation is often easy. No storage of large matrices is needed. In the Lanczos method, only two vectors and the relevant information on **B** to build the sparse matrix-vector product **Bv** need to be used at each step if only the eigenvalues are required.

In theory, the Lanczos algorithm reduces the original matrix **B** ($n \times n$) to a sequence of tridiagonal matrices:

$$\mathbf{T}_j = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \beta_j \\ & & & \beta_j & \alpha_j \end{bmatrix}, \ j = 1, \ldots, n$$

For a real symmetric matrix **B**, it involves the construction of a sequence of orthonormal vectors $\mathbf{v}_j$ (for $j = 1, \ldots, n$) from an arbitrary initial vector $\mathbf{v}_1$ by recursively applying the equation:

$$\beta_{j+1}\mathbf{v}_{j+1} = \mathbf{B}\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_j\mathbf{v}_{j-1} \qquad [7]$$

and $\beta_1 = 0$ and $\mathbf{v}_0 = \mathbf{0}$. The vectors $\mathbf{v}_j$ are referred to as the 'Lanczos vectors'. If at each stage, the coefficient α_j is chosen to make $\mathbf{v}_{j+1}$ orthogonal to $\mathbf{v}_j$ and β_{j+1} is chosen to normalize $\mathbf{v}_{j+1}$ to unity, the vectors should form an orthonormal set and the tridiagonal matrix $\mathbf{T}_n$ with diagonal elements α_j and off-diagonal elements β_{j+1} ($j = 1, \ldots, n$) should have the same eigenvalues as **B**. The coefficients α_j and β_{j+1} are determined as follows:

$$\forall j = 1, \ldots, n, \ \alpha_j = \mathbf{v}'_j\mathbf{B}\mathbf{v}_j \quad \text{and} \quad \beta_{j+1} = \mathbf{v}'_{j+1}\mathbf{B}\mathbf{v}_j \qquad [8]$$

The resulting collection of scalar coefficients α_j and β_{j+1} obtained in these orthogonalizations defines the corresponding Lanczos matrices $\mathbf{T}_j$. Paige (1971) showed in his thesis that for the basis Lanczos recursion [7] orthogonalization with respect to only the two most recently generated Lanczos vectors is sufficient to guarantee that each Lanczos vector is orthogonal to all previously generated Lanczos vectors. The idea of the Lanczos procedure is to replace the computation of eigenvalues for the matrix **B** by the computation of those of the simpler

Lanczos matrices $\mathbf{T}_j$ $(j = 1, \ldots, n)$. Cullum and Willoughby (1985) showed that the eigenvalues of the $\mathbf{T}_j$ provide good approximations to some of the eigenvalues of $\mathbf{B}$. So, if the Lanczos recursion is continued until $j = n$, then the eigenvalues of $\mathbf{T}_n$ will be the eigenvalues of $\mathbf{B}$. In this case, $\mathbf{T}_n$ is simply an orthogonal transformation of $\mathbf{B}$ and must have the same eigenvalues as $\mathbf{B}$.

Theoretically, the Lanczos method cannot determine the multiplicity of any multiple eigenvalue of the matrix $\mathbf{B}$. If the matrix $\mathbf{B}$ has only m distinct eigenvalues $(m < n)$ then the Lanczos recursion would terminate after m steps. Additional computations will be needed to determine which of these eigenvalues are multiple also to compute their multiplicities.

The Lanczos method seems attractive for large sparse matrices because the requirements for storage are very small (the values of α_j and β_j for all j). The elements of $\mathbf{v}_j$ and $\mathbf{v}_{j-1}$ for the current value of j must be stored, as well as $\mathbf{B}$, in such a way that the subroutine for computing $\mathbf{Bv}$ from $\mathbf{v}$ takes full advantage of the sparsity of $\mathbf{B}$. The eigenvalues of $\mathbf{B}$ can be found from those of the more easily handled symmetric matrices $\mathbf{T}_j$ $(j = 1, \ldots, n)$, which are then determined by a standard method (eg, the so-called QL algorithm or a bisection method; Martin and Wilkinson, 1968).

However, because of rounding errors, $\mathbf{v}_{j+1}$ will never be exactly orthogonal to $\mathbf{v}_k$ (for all $k \leqslant j - 2$). Consequently, the values of coefficients α_j and β_{j+1} are inaccurate and the nice properties of the $\mathbf{T}_j$ described above are quickly lost. Paige (1971) and Edwards et al (1979) showed that the loss in the orthogonality of the Lanczos vectors occurs essentially when the quantity $[\beta_j(\mathbf{e}_j'\mathbf{x})]$ becomes small ($\mathbf{e}_j$ is the jth base vector, ie, all components are equal to zero except for the jth one which is equal to one, and $\mathbf{x}$ is an arbitrary vector). A first approach to correct the problem consists of reorthogonalizing all or some of the vectors $\mathbf{v}_j$ at each iteration, but the costs of this operation and the computer storage required for all $\mathbf{v}_j$ can be extremely large. Another approach is not to force the orthogonality of the Lanczos vectors by reorthogonalizing but to work directly with the basic Lanczos recursion, accepting the losses in orthogonality and then unravelling the effects of these losses. Because of this failure of orthogonality, the process does not terminate after n steps but can continue indefinitely to produce a tridiagonal matrix of any desired size. Paige (1971) showed that if the tridiagonal matrix is truncated after a number of iterative steps k much greater than n, the resulting tridiagonal matrix $\mathbf{T}_k$ has a group of eigenvalues very close to the correct eigenvalues for each eigenvalue of the original matrix. He also showed that rounding errors only delayed convergence but did not stop it. Indeed, the eigenvalues of the Lanczos matrices are either 'good' eigenvalues, which are true approximations to the eigenvalues of the matrix $\mathbf{B}$, or 'extra' or 'spurious' eigenvalues, which are caused by the losses in orthogonality. Two types of 'spurious' eigenvalues can be distinguished. Type one is a less accurate copy or 'ghost' copy of the good eigenvalue; type two is genuinely spurious. The main difficulty is to find out which of these eigenvalues correspond to eigenvalues of the original matrix $\mathbf{B}$. For that, the inverse iteration method and the corresponding Rayleigh quotient (Wilkinson, 1965; Chatelin, 1988) are used. The identification test which separates the bad eigenvalues from the good ones rests upon certain relationships between the eigenvalues of the Lanczos tridiagonal matrices $\mathbf{T}_j$ and the eigenvalues of the submatrices $\widehat{\mathbf{T}}_j$ obtained by deleting the first row and column of the matrix $\mathbf{T}_j$. Any eigenvalue of $\mathbf{T}_j$, which is also an eigenvalue of the corresponding $\widehat{\mathbf{T}}_j$ matrix, is labelled as 'spurious' and is discarded from the list of computed eigenvalues. All remaining eigenvalues, including all numerically multiple ones, are accepted and labelled as 'good'. So one can directly identify those eigenvalues which are spurious. In the Lanczos procedure, numerically multiple eigenvalues, which differ from each other by less than a user-specified relative tolerance parameter, are accepted as accurate approximations of eigenvalues of the original matrix **B** and the others (simple eigenvalues) may or may not have converged. This is checked by computing error estimates (only on the resulting single isolated eigenvalues). These error estimates are obtained by calculating the product of the kth component (k being the size of Lanczos matrix considered) of the corresponding Lanczos matrix eigenvector **u** by β_{k+1} (Cullum and Willoughby, 1985).

Therefore, in order to apply this technique, it is necessary to first choose a value k for the size of the Lanczos matrix (often greater than the size of the **B** matrix if all eigenvalues are required). Paige (1971) and Cullum and Willoughby (1985) showed that the primary factor determining whether or not it is feasible to compute large numbers of eigenvalues for a given k is the gap ratio (the ratio of the largest gap between two adjacent eigenvalues to the smallest such gap). The smaller this ratio, the easier it is to compute all the eigenvalues of the original matrix. The larger this ratio, the larger the size of the Lanczos matrix required to obtain all eigenvalues of the **B** matrix. Cullum and Willoughby (1985) showed that reasonably well-separated eigenvalues on the extremes of the spectrum of **B** appear as eigenvalues of the Lanczos matrices for relatively small size of the Lanczos matrix. Therefore this method can also be applied to efficiently find the extreme eigenvalues of a sparse matrix, which are required for example in finding optimal relaxation parameters, when iterative successive relaxation methods are used to solve linear systems (Golub and Van Loan, 1983). Since there is no reorthogonalization, eigenvalues which have converged by a given of the Lanczos matrix may begin to replicate as the size of the Lanczos matrix is increased further.

The Lanczos algorithm does not give rules for determining how large the Lanczos matrix must be in order to compute all the eigenvalues. Paige (1971) showed that it takes more than n steps (generally between $2n$ and $8n$ are needed) to compute all eigenvalues of the spectrum. The stopping criterion cannot be determined beforehand.

A method to determine the multiplicities of multiple eigenvalues

The Lanczos procedure cannot directly determine the multiplicities of the computed eigenvalues as eigenvalues of **B**. Unfortunately, the multiplicity of an eigenvalue of a Lanczos matrix has no relationship with its multiplicity in the original matrix **B**. Good eigenvalues may replicate many times as eigenvalues of a Lanczos matrix but be only single eigenvalues of the original matrix **B**. In fact, numerically multiple eigenvalues are accepted as converged approximations to eigenvalues of **B**.

Cullum (personal communication) proposed an approach to determine and compute the multiplicities of multiple eigenvalues of the matrix **B**, but it requires the computation of the eigenvalues of two or more associated matrices, ie, the algorithm presented previously has to be applied several times. The different matrices required are simple modifications of the original **B** matrix. The approach of Cullum is based upon the following property.

If **B** is a real symmetric matrix and λ is an eigenvalue of **B** with multiplicity p ($p \geqslant 2$) then λ will also be an eigenvalue of the matrix obtained from **B** by adding a symmetric rank-one matrix:

$$\overline{\mathbf{B}} = \mathbf{B} + v_1 \mathbf{v}_1 \mathbf{v}_1' \qquad [9]$$

where $\mathbf{v}_1$ is the starting vector used for **B** in the Lanczos algorithm and v_1 is an arbitrary scalar. Theoretically, if the Lanczos procedure is applied to $\overline{\mathbf{B}}$ and with the same $\mathbf{v}_1$ as starting vector, then the tridiagonal matrices $\overline{\mathbf{T}}_j$ generated for $\overline{\mathbf{B}}$ would be related to those generated for **B**:

$$\overline{\mathbf{T}}_j = \mathbf{T}_j + v_1 \mathbf{e}_1 \mathbf{e}_1' \qquad [10]$$

One could continue this approach to determine the specific multiplicities of each of these multiple eigenvalues by considering the matrix: $\overline{\overline{\mathbf{B}}} = \overline{\mathbf{B}} + v_2 \mathbf{v}_2 \mathbf{v}_2'$, where $\mathbf{v}_2$ is the second Lanczos vector generated for **B** or any vector orthogonal to $\mathbf{v}_1$, and v_2 is a scalar which can be equal to v_1. Any eigenvalue of **B** that has multiplicity greater than two will be in the spectrum of $\overline{\overline{\mathbf{B}}}$, and those with multiplicity equal to two will be in the spectrum of $\overline{\mathbf{B}}$ and not in the spectrum of $\overline{\overline{\mathbf{B}}}$. Thus, the procedure could continue with successive transformations of the **B** matrix until a matrix is obtained with no eigenvalues corresponding to eigenvalues of **B**. This approach seems attractive if the multiplicities of multiple eigenvalues of the matrix **B** are small. If this is not the case, this operation will be expensive, because the algorithm must be applied as many times as the largest multiplicities of multiple eigenvalues.

We present here another approach which can be used when the number of multiple eigenvalues of the **B** matrix is small but their multiplicities are large. The above procedure is applied only once to determine which eigenvalues are multiple and then the following expressions are used to compute the multiplicities of each multiple eigenvalue. First, one makes use of the fact that the total number of eigenvalues of **B** is equal to its dimension:

$$\dim[\mathbf{B}] = N = N_{\mathrm{s}} + \sum_{i=1}^{r} m_i \qquad [11]$$

where N is the dimension of matrix **B**, N_{s} is the number of single nonzero eigenvalues, r is the number of multiple nonzero eigenvalues and m_i is the multiplicity of the ith multiple eigenvalue γ_i.

It is also known that:

$$\mathrm{tr}[\mathbf{B}] = \sum_{j=1}^{N_{\mathrm{s}}} \lambda_j + \sum_{i=1}^{r} m_i \gamma_i \qquad [12]$$

where the λ_j are the simple eigenvalues.

$$\mathrm{tr}[\mathbf{B}^2] = \sum_{j=1}^{N_s} \lambda_j^2 + \sum_{i=1}^{r} m_i \gamma_i^2 \qquad [13]$$

Finally, the m_i are obtained solving the (possibly overdetermined) system of three equations, for example, using integer linear programming subroutines (the m_i must be integers).

The traces of $\mathbf{B}$ and $\mathbf{B}^2$ are simply obtained by using the subroutine of the sparse matrix-vector multiplications which will be presented in the next part in a particular setting. The matrix $\mathbf{B}$ need not be stored explicitly. Only two vectors are needed to compute these traces:

$$\mathrm{tr}[\mathbf{B}] = \sum_{i=1}^{n} b_{ii} = \sum_{i=1}^{n} (\mathbf{Be}_i)_i \qquad [14]$$

where the $\mathbf{b}_{ii}$ are the diagonal elements of the $\mathbf{B}$ matrix, $\mathbf{e}_i$ is the ith base vector, and $(\mathbf{Be}_i)_i$ represents the ith element of the vector $(\mathbf{Be}_i)$.

$$\mathrm{tr}[\mathbf{B}^2] = \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} ((\mathbf{Be}_i)_j)^2 \qquad [15]$$

The validity of the approach proposed here to determine multiplicities was verified on several examples for matrices of moderate size (up to $n = 1\,000$). For such matrices, a regular approach (for example, routine F02AAF of the NAG Fortran Library) working on dense matrices stored in the core can be used to compute all eigenvalues and their multiplicities. It was found that Cullum and Willoughby's method applied using a Lanczos matrix of size $k = 2n$ and computing multiplicities from equations [11], [14] and [15] as proposed above, leads to exactly the same results as the regular approach.

Sparse computation of the matrix vector product Bv

Here we illustrate the previous method and its use with sparse matrix techniques to compute [2] and/or [3] in the context of a simple animal model with one fixed effect. In a genetic context, a typical mixed animal model is characterized by:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Za} + \mathbf{e} \qquad [16]$$

with $$\mathrm{E}\begin{bmatrix}\mathbf{a}\\ \mathbf{e}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} \quad \text{and} \quad \mathrm{Var}\begin{bmatrix}\mathbf{a}\\ \mathbf{e}\end{bmatrix} = \begin{bmatrix}\mathbf{A}\sigma_a^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\sigma_e^2\end{bmatrix}$$

where:
$\mathbf{y} =$ data vector,
$\boldsymbol{\beta} =$ vector of fixed effect,
$\mathbf{a} =$ vector of random additive genetic effects, such that $\mathbf{a} \sim N(\mathbf{0}, \mathbf{A}\sigma_a^2)$,

$\mathbf{X}, \mathbf{Z} =$ known incidence matrices associated with vectors $\boldsymbol{\beta}$ and $\mathbf{a}$ respectively,
$\mathbf{e} =$ vector of random residuals, such that $\mathbf{e} \sim N(\mathbf{0}, \mathbf{I}\sigma_e^2)$,
$\mathbf{A} =$ numerator relationship matrix among animals (Henderson, 1975).

The mixed-model equations (MME) of Henderson (1973) are:

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'Z} \\ \mathbf{Z'X} & \mathbf{Z'Z} + \alpha\mathbf{A}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'Y} \\ \mathbf{Z'Y} \end{bmatrix}$$

Absorption of the fixed effect equations leads to:

$$[\mathbf{Z'MZ} + \alpha\mathbf{A}^{-1}]\widehat{\mathbf{a}} = \mathbf{Z'MY} \qquad [17]$$

where $\mathbf{M} = \mathbf{I} - \mathbf{X}(\mathbf{X'X})^{-}\mathbf{X'}$ and $\alpha = \sigma_e^2/\sigma_a^2$.

The estimation of parameters σ_a^2 and σ_e^2 by an EM-REML (or a Fisher-scoring) procedure involves the computation of the trace given in [5] (or [5] and [6] for Fisher-scoring). The matrix $\mathbf{B}$ considered in the Lanczos method corresponds to $\mathbf{L'Z'MZL}$ here. We will assume that, before the computation of any matrix vector product $\mathbf{Bv} = \mathbf{u}$ (where the vector $\mathbf{v}$ is arbitrary), a pedigree file and a data file have been read and that their information has been stored into three vectors of size n, where n is the size of $\mathbf{A}$; the sire, dam and level of the fixed effect of each animal are stored in s_i, d_i and h_i, respectively ($h_i = 0$ if the animal has no record). Progeny must precede parents. Simultaneously, the number of observations in each level of the fixed effect is cumulated in a vector of size n_f. Note that $\mathbf{u} = \mathbf{Bv} = \mathbf{L'Z'MZLv} = \mathbf{L'Z'ZLv} - \mathbf{L'Z'X}(\mathbf{X'X})^{-}\mathbf{X'ZLv}$. The sparse computation of $\mathbf{Bv} = \mathbf{u}$ involves the following loops:

(1) compute $\mathbf{w} = \mathbf{Lv}$ and $\mathbf{t} = (\mathbf{X'Z})\mathbf{Lv}$
(2) compute $\mathbf{f} = (\mathbf{X'X})^{-}\mathbf{t}$
(3) compute $\mathbf{u} = \mathbf{L'Z'}(\mathbf{Zw} - \mathbf{Xf}) = \mathbf{L'r}$

The computation of $\mathbf{w} = \mathbf{Lv}$ in [1] and $\mathbf{u} = \mathbf{L'r}$ in [3] is performed by solving the sparse triangular systems $\mathbf{L}^{-1}\mathbf{w} = \mathbf{v}$ and $\mathbf{L}^{-\mathbf{T}}\mathbf{u} = \mathbf{r}$. Each line i of $\mathbf{L}^{-1}$ and each column of $\mathbf{L}^{-\mathbf{T}}$ contains at most three nonzero elements which can be easily identified (the diagonal element + the elements in columns (or lines) s_i and d_i). Vector $\mathbf{t}$ is obtained by simply cumulating elements of $\mathbf{w}$ corresponding to animals with records. Similarly, elements of $\mathbf{r}$ are equal to the difference between the elements of $\mathbf{w}$ and the appropriate elements of $\mathbf{f}$ for animals with records and to zero for the others. Note that only five vectors of size n ($\mathbf{s}, \mathbf{d}, \mathbf{h}, \mathbf{u}$ and $\mathbf{v}$) and two of size n_f ($\mathbf{f}$ and $\mathbf{t}$) are required.

EXAMPLE

To illustrate the method, a data set including the type scores of 9 686 dairy cows and their ancestors (21 269 animals in total) for 18 type traits was created. To estimate the genetic parameters of these type traits, the animal model [16] was used assuming precorrection of the data for age at calving and stage of lactation, and using a month-herd-class classifier effect as a unique fixed effect (292 levels). A canonical transformation of the data can be applied (eg, Meyer, 1985) because all traits are analyzed according to the same model with equal design matrices. However, it was considered that the repeated computations of expressions [2] and/or [3] for matrices of size $n = 21\,269$ could be advantageously replaced by a unique computation of all eigenvalues of the matrix $\mathbf{B} = \mathbf{L}'\mathbf{Z}'\mathbf{M}\mathbf{Z}\mathbf{L}$ followed by the repeated use of equalities [5] and [6] in a Fisher-scoring REML iterative procedure.

The main programs were supplied by Cullum and the sparse computation of the matrix vector product $\mathbf{Bv}$ was done according to the strategy described above on an IBM Risc 6000/590 computer. Tridiagonal matrices $\mathbf{T}_k$ with $k = 2n$, $4n$, $6n$, $8n$ and $10n$ were computed using the Lanczos recursion [7]. The procedure was applied twice, once on $\mathbf{B}$ and once on $\overline{\mathbf{B}}$ [9] in order to detect multiple eigenvalues, then their multiplicities were calculated using the relationships [11–13] and an integer linear programming subroutine.

REML estimates of genetic and residual (co)variances were obtained repeatedly using the eigenvalues computed from Lanczos matrices of increasing size in [5] and [6]. The quadratic forms required at each Fisher-scoring iteration were calculated by iteratively solving the mixed-model equations of a multiple-trait best linear unbiased prediction (BLUP) animal model, after canonical transformation and with starting values equal to the solutions of the previous system of equations. Estimates of asymptotic standard errors of (co)variance parameters were available from the inverse of the information matrix at convergence of the Fisher-scoring iterative procedure. Estimates of heritabilities, genetic and residual correlations and the corresponding asymptotic errors were also compared.

RESULTS

Table I shows the number of distinct eigenvalues computed and the calculated multiplicities of the four multiple eigenvalues (0, 0.50, 0.6875, 0.75) encountered, and the CPU time required for the different values of k ($2n$, $4n$, $6n$, $8n$ and $10n$). CPU time mainly consisted of two parts: the time required for the computation of the Lanczos matrices $\mathbf{T}_k$ increased linearly with k. The computing cost for the determination of the eigenvalues of the tridiagonal matrices $\mathbf{T}_k$ increased quadratically with k and was always much larger than the calculation of $\mathbf{T}_k$, for the simple model with only one fixed effect considered here.

Obviously, very large Lanczos matrices must be used in order to detect all eigenvalues: 31 new distinct eigenvalues were detected when the size of the Lanczos matrix was increased from $8n$ to $10n$. For $k = 10n$, all the eigenvalues found had a good precision (at least 10^{-5} with very few precisions worse than 10^{-10}) but there is a possibility that other eigenvalues may still have been kept undetected. As a result, the multiplicities of the multiple eigenvalues obtained using [11–13] differ when the size of the Lanczos matrix increases. However, a close look at the eigenvalues shows that all distinct eigenvalues have been found with the Lanczos matrix $\mathbf{T}_k$ for $k = 4n$ with the rare exception of some of those located in the intervals [0.72; 0.75] and [0.84; 0.87]. As k increases, these intervals become smaller: for $k = 6n$, these intervals are [0.73; 0.75] and [0.86; 0,87] and for $k = 8n$ [0.74; 0.75] only. The difficulty of detecting all eigenvalues in clusters of very close eigenvalues was clearly identified as a drawback of their method by Cullum and Willoughby (1985).

Table I. Computing time and characteristics of the eigenvalues obtained from Lanczos matrices of increasing size.

*Size of the Lanczos matrix (*k*)*	*2*n	*4*n	*6*n	*8*n	*10*n
CPU time[a]	00:09:49	1.98	2.91	3.95	4.60
CPU time[b]	00:32:29	3.84	8.48	14.87	25.50
Number of distinct eigenvalues	2 086	2 379	2 494	2 560	2 591
Multiplicity of $\lambda = 0$	11 875	11 875	11 875	11 875	11 875
Multiplicity of $\lambda = 0.5$	153	93	83	83	83
Multiplicity of $\lambda = 0.6875$	5 947	6 425	6 495	6 495	6 494
Multiplicity of $\lambda = 0.75$	1 212	501	326	260	230

[a] CPU time for computing the tridiagonal matrix $\mathbf{T}_k$ (in h:min:s for $k = 2n$ and in relative value for $k = 4n$, $6n$, $8n$ and $10n$). [b] CPU time for determining all eigenvalues of the tridiagonal matrix $\mathbf{T}_k$ (in h:min:s for $k = 2n$ and in relative value for $k = 4n$, $6n$, $8n$ and $10n$).

Table II presents the value of expressions [2] and [3] obtained with $k = 10n$ and the relative precision of expressions [2] and [3] from the eigenvalues and the multiplicities obtained in table I for three different values of α (99, 4 and 1/3) corresponding to an extreme range of heritabilities ($h^2 = 0.01$, 0.20 and 0.75). The eigenvalues obtained for $k = 10n$ are assumed to be true values. The striking result is that, whatever the value of α considered and although some distinct eigenvalues are undetected and the multiplicities of the multiple eigenvalues are incorrect, the traces considered are very well approximated when a Lanczos matrix of size $k = 2n$ is used and are virtually exact when $k \geqslant 4n$.

Table II. Relative error for traces computed using eigenvalues obtained from Lanczos matrices of increasing size.

	$tr[\boldsymbol{B} + \alpha\boldsymbol{I}]^{-1}$			$tr[\boldsymbol{B} + \alpha\boldsymbol{I}]^{-2}$		
k[a] α	99	4	1/3	99	4	1/3
$2n$	-9×10^{-5}	-9×10^{-5}	-2×10^{-5}	-9×10^{-5}	-8×10^{-5}	2×10^{-5}
$4n$	-3×10^{-8}	-7×10^{-7}	1×10^{-6}	-7×10^{-8}	-1×10^{-6}	3×10^{-6}
$6n$	-2×10^{-8}	-3×10^{-7}	-1×10^{-7}	-3×10^{-8}	-5×10^{-7}	3×10^{-7}
$8n$	-3×10^{-9}	-4×10^{-8}	2×10^{-8}	-5×10^{-9}	-5×10^{-8}	7×10^{-8}
$10n$[b]	213.9391	4 918.8131	44 358.3198	2.1522	1 151.7537	115 330.900

[a] Size of the Lanczos matrix; [b] values of the traces considered as the exact ones.

It is well known that small departures from the true values of the traces can lead to rather large biases in the final REML estimates due to the iterative algorithms used. This explained some disappointing conclusions when approximations of traces were proposed (eg, Boichard et al, 1992). Table III reports some characteristics of the estimates of the genetic parameters calculated using the eigenvalues obtained from Lanczos matrices of different size for the computations of traces like those in [5] and [6]. A Fisher-scoring algorithm was used and stopped after ten iterations for

Table III. Errors on (co)variance estimates when the traces required in the Fisher-scoring algorithm are computed using eigenvalues obtained from Lanczos matrices of increasing size.

Parameter	*Criterion*[a]	*Size of the Lanczos matrix*			
		*2*n	*4*n	*6*n	*8*n
Heritability	Mean $\left\lvert \widehat{h}_i^2 - h_i^{*2} \right\rvert$	2×10^{-4}	3×10^{-5}	3×10^{-5}	4×10^{-5}
Heritability	Max $\left\lvert \widehat{h}_i^2 - h_i^{*2} \right\rvert$	6×10^{-4}	9×10^{-5}	6×10^{-5}	6×10^{-5}
Heritability	Max $\left\lvert \widehat{h}_i^2 - h_i^{*2} \right\rvert \Big/ h_i^{*2}$	1.1×10^{-3}	2.6×10^{-4}	1.6×10^{-4}	1.7×10^{-4}
Genetic variance	Max $\left\lvert \widehat{\sigma}_i^2 - \sigma_i^{*2} \right\rvert / \sigma_i^{*2}$	1.1×10^{-3}	1.9×10^{-4}	1.9×10^{-4}	2.1×10^{-4}
Asymptotic standard error of the heritability	Max $\left\lvert \widehat{\sigma} \left(\widehat{h}_i^2 \right) - \sigma^* \left(h_i^{*2} \right) \right\rvert \Big/ \sigma^* \left(h_i^{*2} \right)$	9×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Genetic correlation	Max $\left\lvert \widehat{\rho}_{ij} - \rho_{ij}^* \right\rvert$	3×10^{-4}	4×10^{-5}	3×10^{-5}	3×10^{-5}

[a] '∧' refers to the results obtained with the size of the Lanczos matrix indicated; '*' refers to the results obtained with a Lanczos matrix of size $10n$ (regarded as exact results).

each run, although convergence was practically achieved on all parameters after five or six iterations. Each complete run for 18 traits treated simultaneously took about 4 min of CPU time, mainly devoted to the solution of the mixed-model equations. Estimates of parameters (variances, heritabilities and correlations) were virtually identical regardless of the value of k. This is obviously a consequence of the good quality of the approximation of the traces reported in table II.

DISCUSSION AND CONCLUSION

This example clearly shows the feasibility of the computation of all eigenvalues for very large sparse matrices. The main difficulty is to determine the size k of the tridiagonal matrix $\mathbf{T}_k$ to be used in practice. Cullum and Willoughby's approach has two main limitations: a) eigenvalues in small dense clusters are difficult to find; and b) there is no really satisfying way to compute the multiplicities of multiple eigenvalues when these multiplicities are very large. However, in the particular case when eigenvalues are used only to calculate traces, these two drawbacks annihilate each other, at least when the approach proposed here to determine the multiplicities is chosen and when the number of multiple eigenvalues is small; the use of incorrect multiplicities compensates for undetected eigenvalues. Therefore, there is no need to find all eigenvalues to have an excellent approximation of the quantities required in first- and second-order REML algorithms.

Our main objective was to show that the use of the simple expressions [5] and [6] is not limited to small, dense matrices. Many new directions of research are opened. The efficient computation of the matrix product $\mathbf{Bv}$ for any vector $\mathbf{v}$ in more complex models is a key point for extending this approach to other situations. For example, we were able to compute $\mathbf{Bv} = \mathbf{L'Z'MZLv}$ when another fixed effect (a group of unknown parent effect) was added to the model used in our example. CPU time for the Lanczos recursion was doubled but the computation of the eigenvalues of the tridiagonal matrices remained by far the most time-consuming part. Other promising techniques like the 'divide and conquer' approach (Dongarra and Sorensen, 1987) could be much more attractive than the traditional QL method used here. These techniques designed to take full advantage of parallel computations, when these are possible, may still significantly reduce the time required for the diagonalization of the Lanczos matrices when used in serial mode (Sorensen, personal communication).

The value of knowledge of the eigenvalues of large sparse matrices is not limited to REML estimation. It appears for example in the analysis of the contributions of different lines to the estimation of genetic parameters from selection experiments (Thompson and Atkins, 1994). Sometimes, only extreme eigenvalues are needed, eg, in the computation of the optimal relaxation factors in iterative algorithms for solving linear systems (Golub and Van Loan, 1983). In the Lanczos procedure, information about $\mathbf{B}$s extreme eigenvalues tends to emerge long before the tridiagonalization is complete (for $k < n$). In our example, the largest eigenvalue was detected with a value of k as low as 5.

In situations where the knowlege of all eigenvalues is necessary, the problem of the multiplicities of multiple eigenvalues may be tackled differently. The three nonzero eigenvalues observed in our example (0.5; 0.6875; 0.75) correspond to groups of animals with similar characteristics (full-sibs, same sire and maternal grandsire, half-sibs) as already pointed out by Thompson and Shaw (1990). However, we were not able to determine the expected multiplicities by a careful look at the pedigree file (except for full-sibs). If an efficient algorithm to compute these eigenvalues were available, the computation of eigenvalues using the Lanczos method could be limited to a modified smaller matrix, as suggested by Thompson and Shaw (1990), at least as long as the sparsity of the matrix is not significantly altered.

ACKNOWLEDGMENT

The authors are extremely grateful to JK Cullum (IBM, Yorktown Heights, NY, USA) for providing the Lanczos subroutine, for her valuable comments and for her help in understanding and implementing the Lanczos algorithm in a genetic context. The initial help of A Tabary (IBM France, Paris, France) and some interesting discussions with R Thompson (Roslin Institute, Edinburgh, UK) are also acknowledged.

REFERENCES

Boichard D, Schaeffer LR, Lee AJ (1992) Approximate restricted maximum likelihood and approximate prediction error variance of the mendelian sampling effect. *Genet Sel Evol* 24, 331-343

Chatelin F (1988) *Valeurs propres de matrices.* Masson, Paris

Colleau JJ, Beaumont C, Regaldo D (1989) Restricted maximum likelihood (REML) estimation of genetic parameter for type traits in Normande cattle breed. *Livest Prod Sci* 23, 47-66

Cullum JK, Willoughby RA (1985) Lanczos algorithms for large symmetric eigenvalue computations. Vol I: Theory, Vol II : programs. Birkhäuser Boston Inc, Boston, MA

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Statist Soc B* 39, 1-38

Dempster AP, Selwyn MR, Patel M, Roth AJ (1984) Statistical and computational aspects of mixed model analysis. *Appl Stat* 33, 203-214

Dongarra JJ, Sorensen DC (1987) A fully parallel algorithm for the symmetric eigenvalue problem. *SIAM J Sci Stat Comput* 8, 139-154

Ducrocq V (1993) Genetic parameters for type traits in the French Holstein breed based on a multiple trait animal model. *Livest Prod Sci* 36, 143-156

Edwards JT, Licciardello DC, Thouless DJ (1979) Use of the Lanczos method for finding complete sets of eigenvalues of large sparse matrices. *J Inst Math Appl* 23, 277-283

García-Cortés LA, Moreno C, Varona L, Altarriba J (1992) Variance component estimation by resampling. *J Anim Breed Genet* 109, 358-363

Golub GH, Van Loan CF (1983) *Matrix Computations.* Johns Hopkins Univ Press, Baltimore, MD

Graser HU, Smith SP, Tier B (1987) A derivative-free approach for estimating variance components in animal models by restricted maximum likelihood. *J Anim Sci* 64, 1362-1370

Harville DA, Callanan TP (1990) Computational aspects of likelihood based inference for variance component. In: *Advances in Statistical Methods for Genetic Improvement of Livestock Production* (D Gianola, K Hammond, eds), Springer Verlag, Heidelberg, 136-176

Henderson CR (1973) Sire evaluation and genetic trends. In: *Proceedings of Animal Breeding and Genetics Symposium in Honor of Dr JL Lush.* Am Soc Anim Sci–Am Dairy Sci Assoc, Champaign, IL, 10-41

Henderson CR (1975) Use of all relatives in intraherd prediction of breeding values and producing abilities. *J Dairy Sci* 58, 1910-1916

Henderson CR (1976) A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding value. *Biometrics* 32, 69-83

Lanczos C (1950) An iterative method for the solution of the eigenvalue problem linear differential and integral operators. *J Res Nat Bur Standards* 45, 255-282

Lin CY (1987) Application of singular value decomposition to restricted maximum likelihood estimation of variance components. *J Dairy Sci* 70, 2680-2784

Martin RS, Wilkinson JH (1968) The implicit QL algorithm. *Numer Math* 12, 377-383

Meyer K (1985) Maximum likelihood estimation of variance components for a multivariate mixed model with equal design matrices. *Biometrics* 41, 153-165

Misztal I (1990) Restricted maximum likelihood estimation of variance components in animal model using sparse matrix inversion and a supercomputer. *J Dairy Sci* 73, 163-172

Misztal I, Perez-Enciso M (1993) Sparse matrix inversion for restricted maximum likelihood estimation of variance components by expectation-maximization. *J Dairy Sci* 76, 1479-1483

Misztal I (1994) Comparison of computing properties of derivative and derivative-free algorithms in variance component estimation by REML. *J Anim Breed Genet* 111, 346-355

Paige CC (1971) The computation of eigenvalues and eigenvectors of very large sparse matrices. PhD Thesis, London

Patterson HD, Thompson R (1971) Recovery of interblock information when block sizes are unequal. *Biometrika* 58, 545-554

Smith SP, Graser HU (1986) Estimating variance components in a class of mixed models by restricted maximum likelihood. *J Dairy Sci* 69, 1156-1165

Thallman RM, Taylor JF (1991) An indirect method of computing REML estimates of variance components from large data sets using an animal model. *J Dairy Sci* 74 (Suppl I), 160 (Abstr)

Thompson EA, Shaw RG (1990) Pedigree analysis for quantitative traits: variance components without matrix inversion. *Biometrics* 46, 399-413

Thompson R, Atkins KD (1994) Sources of information for estimating heritability from selection experiments. *Genet Res* 63, 49-55

Wilkinson JH (1965) *The Algebraic Eigenvalue Problem.* Clarendon, Oxford